Quality Assurance and Security: Can They Be Synonymous in Software?

Glenn Johnson

Certification Consultant for (ISC)^2

www.isc2.org

20 (ISC)^2 years of excellence

Overview & Background

- Established in 1989 – Non-profit consortium of information security industry professionals

- Global leaders in certifying and educating information security professionals throughout their careers

- Global standard for information security – (ISC)² CBK®, a compendium of information security topics as certified by ANSI/ISO/IEC Standard 17024

- 65,000+ certified professionals more than 130 countries

What Is Quality Assurance?

Software quality assurance is the system of repeatable and measurable processes and procedures used to assure that the software product meets its requirements. The system involves planning, measuring and monitoring developmental activities performed by others.

vs.

What Is Software Security?

Security is a distinct property of a software system or application. It is composed of Confidentiality, Integrity, Availability, Authenticity, and other related attributes*.

- Software Security vs. Secure Software
  - Secure software can be delivered by rigorously applying all the techniques of a software security plan
- Software Security vs. Secure Coding
  - Secure coding is one aspect of an overall software security plan

*Definition derived from description provided in Software Assurance BoK from DHS.

What Is Software Security?

Software Security vs. Software Quality

- High-quality software can also be insecure
- Security requires specialized skills

* (e.g. Accountability, Anonymity, Non-repudiation, and others.)

vs.

*Definition derived from description provided in Software Assurance BoK from DHS.

The Changing Landscape of Security

- Over 70% of security vulnerabilities exist at the application layer*
- Perimeter protection no longer sufficient – data compromise is the issue
- More incidents of data loss could result in greater government oversight and regulation
- 2008 (ISC)² Global Information Security Workforce Study report found significant costs result from data breaches
    - US $50 to $200 per record lost (not including reputation damage and loss of trust)

*Gartner Group, 2005

Business Impact

The ramifications of insecure software go beyond mere technology issues; there is also a definite business impact.

Not having secure software can lead to:

- Financial loss
- Bad publicity
- Investigations and litigation
- Liability (personal and corporate)
- Reputation damage
- Loss of brand, confidence and trust among customers, partners, shareholders and stakeholders

Insecure Software: Policy Problem

Software developers are driven to deliver functionality within deadline and scope constraints because of:

- Lack of time
- Expense
- Limited personnel resources
- Rush to market
- Lack of awareness of the value of security

Insecure Software: Process Problem

Developers have little appreciation for basic security tenets

- Protection from disclosure (confidentiality)
- Protection from alteration (integrity)
- Protection from destruction (availability)
- Validating who is making the request (authentication)
- What rights and privileges does the requestor have (authorization)
- The ability to build historical evidence (auditing) and the management of configuration, sessions and exceptions
- If they are aware of the principles, do they understand the implementation practices?

Insecure Software: People Problem

Three primary conditions create information security vulnerabilities in enterprise software applications:

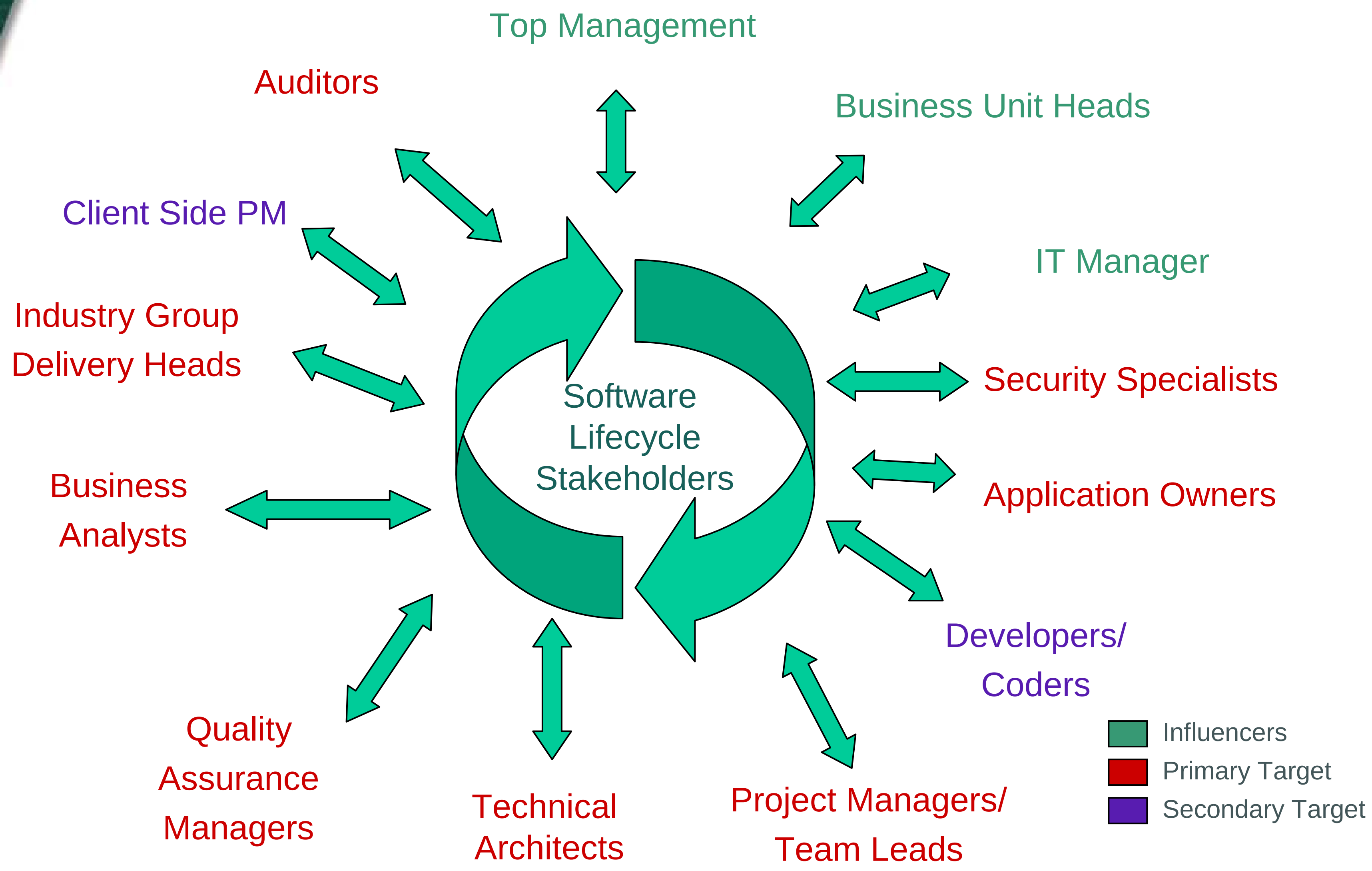
- Inexperienced developers writing code
- Experienced developers writing code with inadequate training in best practices for security
- Designers and managers failing to include security considerations prior to development.
- Influencers not understanding information security issues as they pertain to the secure software lifecycle

Secure Software Domains

Based on (ISC)² CSSLP CBK

- Secure Software Concepts
- Secure Software Requirements
- Secure Software Design
- Secure Software Implementation/Coding
- Secure Software Testing
- Software Acceptance
- Software Deployment, Operations, Maintenance, and Disposal

Who Are the Stakeholders?



Software Lifecycle Stakeholders

Top Management

Auditors

Business Unit Heads

Client Side PM

IT Manager

Industry Group Delivery Heads

Security Specialists

Application Owners

Business Analysts

Developers/ Coders

Quality Assurance Managers

Technical Architects

Project Managers/ Team Leads

🟩 Influencers

🟥 Primary Target

🟪 Secondary Target

Secure Software Concepts from the CSSLP®

- Confidentiality, Integrity, Availability Authentication, Authorization, and Auditing
- Security Design Principles
- Risk Management (e.g., vulnerabilities, threats and controls)
- Regulations, Privacy, and Compliance
- Software Architecture (e.g., layers)
- Software Development Methodologies
- Legal (e.g., Copyright, IP and trademark)
- Standards (e.g., ISO 2700x, OWASP)
- Security Models (e.g., Bell-LaPadula, Clark-Wilson & Brewer-Nash)
- Trusted Computing (e.g., TPM, TCB)
- Acquisition (e.g., contracts, SLAs and specifications)

THANK YOU!

Special thanks to (ISC)^2 advisor and Microsoft professional Jim Molini. See his blog at www.codeguard.org/blog

CONTACT ME:
GLENN JOHNSON
GJOHNSON@ISC2.ORG